# Feature-Based Dissimilarity Space Classification

Robert P.W. Duin[1], Marco Loog[1], Elżbieta Pękalska[2], and David M.J. Tax[1]

[1] Faculty of Electrical Engineering, Mathematics and Computer Sciences,
Delft University of Technology, The Netherlands
r.duin@ieee.org, m.loog@tudelft.nl, d.m.j.tax@tudelft.org

[2] School of Computer Science, University of Manchester, United Kingdom
pekalska@cs.man.ac.uk

**Abstract.** General dissimilarity-based learning approaches have been proposed for dissimilarity data sets [1,2]. They often arise in problems in which direct comparisons of objects are made by computing pairwise distances between images, spectra, graphs or strings.

Dissimilarity-based classifiers can also be defined in vector spaces [3]. A large comparative study has not been undertaken so far. This paper compares dissimilarity-based classifiers with traditional feature-based classifiers, including linear and nonlinear SVMs, in the context of the ICPR 2010 Classifier Domains of Competence contest. It is concluded that the feature-based dissimilarity space classification performs similar or better than the linear and nonlinear SVMs, as averaged over all 301 datasets of the contest and in a large subset of its datasets. This indicates that these classifiers have their own domain of competence.

## 1 Introduction

Pairwise dissimilarities constitute a natural way to represent objects. They may even be judged as more fundamental than features [4]. Vector spaces defined by pairwise dissimilarities computed between objects like images, spectra and time signals offer an interesting way to bridge the gap between the structural and statistical approaches to pattern recognition [1,2]. Structural descriptions may be used by the domain experts to express their specific background knowledge [5,6]. Such descriptions often rely on graphs, strings, or normalized versions of the raw measurements, while maintaining the object connectivity in space, frequency or time. A well chosen dissimilarity measure is used to compare objects to a fixed set of representation objects. Such dissimilarity vectors construct a vector space, the so-called dissimilarity space. Traditional classifiers, designed for feature spaces, can be constructed in the dissimilarity space.

This dissimilarity approach may also be used on top of a feature representation [3]. It offers thereby an alternative to kernel approaches based on similarities. Dissimilarity measures are more general than kernels. The later have to obey the Mercer condition so that the implicit construction of classifiers, such as Support Vector Machine (SVM), can be possible in the related kernel spaces. The dissimilarity approach has the advantage that any measure can be used as well as any classifier that works in vector spaces.

It is the purpose of this paper to present a large scale comparison between traditional classifiers built in a feature vector space and some appropriate classifiers built in the

D. Ünay, Z. Çataltepe, and S. Aksoy (Eds.): ICPR 2010, LNCS 6388, pp. 46–55, 2010.

dissimilarity space defined over the original feature space. This dissimilarity space is built by the Euclidean distances to the set of chosen representation objects. The dimension of the space equals the size of the representation set. Various studies are available on the selection of this set out of the training set [7,8] and classification results depend on such a selection procedure. To simplify our experiments we will restrict ourselves to representation sets that are equal to the training set. It means that the number of training objects is identical to the dimension of the space. Consequently, we focus on classifiers in the dissimilarity space that can handle this situation.

## 2 The Dissimilarity Representation

The dissimilarity representation has extensively been discussed, e.g. in [1] or [9], so we will only focus here on some aspects that are essential for this paper.

Traditionally, dissimilarity measures were optimized for the performance of the nearest neighbor rule. In addition, they were also widely used in hierarchical cluster analysis. Later, the resulting dissimilarity matrices served for the construction of vector spaces and the computation of classifiers. Only more recently proximity measures have been designed for classifiers that are more general than the nearest neighbor rule. These are usually similarities and kernels (but not dissimilarities) used in combination with SVMs. So, research on the design of dissimilarity measures such that they fit to a wide range of classifiers is still in an early stage. In this paper we focus on the Euclidean distance derived in the original feature space. Most traditional feature-based classifiers use the Euclidean distance measure in one way or the other as well. It is our purpose to investigate for which datasets such classifiers can be improved by transforming the feature space into a dissimilarity space, both relying on the same Euclidean distance.

Given a set of pairwise dissimilarities between all training objects, the so-called dissimilarity matrix, we studied two ways of constructing a vector space [1]: the postulation of a dissimilarity space and a (pseudo-Euclidean) embedded space. Because the dissimilarity matrix we compute here is the Euclidean distance matrix in the feature space, the resulting embedded space is the original feature space. Therefore, we will just deal with the dissimilarity space, which is introduced now more formally.

### 2.1 Dissimilarity Space

Let $\mathscr{X} = \{o_1, \dots, o_n\}$ be a training set of objects $o_i$, $i = 1, \dots, n$. In general, these are not necessarily vectors but can also be real world objects or e.g. images or time signals. Given a dissimilarity function and/or dissimilarity data, we define a data-dependent mapping $D(\cdot, R) : \mathscr{X} \to \mathbb{R}^k$ from $\mathscr{X}$ to the so-called *dissimilarity space* [10,11,12]. The $k$-element set $R$ consists of objects that are representative for the problem. This set, the representation or prototype set, may be a subset of $\mathscr{X}$. In the dissimilarity space each dimension $D(\cdot, p_i)$ describes a dissimilarity to a prototype $p_i$ from $R$. Here, we will choose $R := \mathscr{X}$. As a result, every object is described by an $n$-dimensional dissimilarity vector $D(o, \mathscr{X}) = [d(o, o_1) \ \dots \ d(o, o_n)]^T$, which is a row of the given dissimilarity matrix $D$. The resulting vector space is endowed with the traditional inner product and the Euclidean metric. Since we have $n$ training objects in an $n$-dimensional space, a classifier such as SVM is needed to handle this situation. Other solutions such as dimension reduction by PCA or prototype selection are not considered here with one exception, i.e. the use of a representation set, randomly selected out of the training set and consisting of 20% of the training objects. We will then compare classifiers built in complete dissimilarity spaces with classifiers built in the reduced spaces (defined over smaller representation sets), yielding five times as many objects as dimensions.

Since the dissimilarity space is defined by the Euclidean distance between the objects in the feature space and, in addition, we also use Euclidean distance over the dissimilarity space, it can easily be shown that asymptotically (for growing representation sets and training sets) the nearest neighbors in the dissimilarity space are identical to the nearest neighbors in the feature space. This does not hold, however, for finite sets. This is an advantage in case of noisy features: nearest neighbors in the dissimilarity space are more reliable than in the feature space because noise is reduced by averaging in the process of computing distances.

### 2.2 Feature-Based Dissimilarity Space Classification

Feature-based Dissimilarity Space (FDS) classification is now defined as follows:

1. Determine all pairwise distances as an $n \times n$ dissimilarity matrix $D$ between the $n$ objects in the training set $\mathscr{X} = \{o_1, \ldots, o_n\}$. $D_{ij}$ is the Euclidean distance between the $i$-th and $j$-th objects.
2. Define the dissimilarity space as a Euclidean vector space $X$ by $X = D$. Hereby, an $i$-th object is represented by a dissimilarity vector of the $D_{ij}$-values, $j = 1, \ldots, n$.
3. Train classifiers on the $n$ training objects represented in the $n$-dimensional dissimilarity space.
4. Test objects are represented in the dissimilarity space by their Euclidean distances to the objects in the training set and applied to the trained classifier in this space.

Traditional classifiers can now be used as FDS classifiers. We will study three classifiers here: the (pseudo-)Fisher linear discriminant, the logistic classifier and linear SVM. Concerning computational effort, the dimensionality is increased to the feature size $n$. In particular, the Fisher discriminant may suffer from this as it relies on the inverse of an $n \times n$ covariance matrix.

In order to illustrate some properties of FDS classification on a two-dimensional spiral dataset we compare two classifiers: an optimized radial basis SVM computed in the feature space (implicitly in the radial basis kernel space) and a Fisher discriminant in the dissimilarity space; see Figure 1. The later classifier is overtrained and results in a hyperplane having exactly the same distances to all objects in the training set. This works out such that in the feature space the distances to the most neighboring objects are still about equal on all positions of the spiral. This does not hold for SVM whose constant kernel is too narrow on the outside of the spiral and too large in the center.

## 3 Observations on the Datasets

We use the ICPR2010 Landscape contest for a systematic comparison with a set of other classifiers of the 301 contest datasets. All but one datasets are just two class problems.

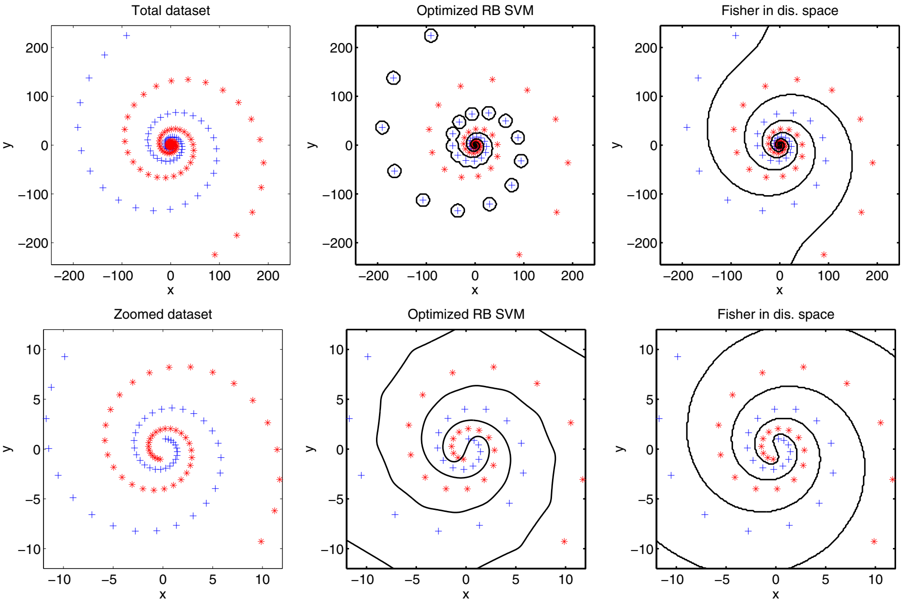

**Fig. 1.** A spiral example with 100 objects per class. Top row shows the complete data sets, while bottom row presents the zoom of the spiral center. 50 objects per class, systematically sampled, are used for training. The middle column shows the training set and the SVM with an optimized radial basis function; 17 out of 100 test objects are erroneously classified. The right column shows the Fisher linear discriminant (without regularization) computed in the dissimilarity space derived from the Euclidean distances. All test objects are correctly classified.

They are either 8-dimensional or 20-dimensional and the class sizes vary between 5 and 938. The largest dataset has 20 classes in 20 dimensions and has almost 10000 objects. We observe that a small amount of artificial noise has been added in a number of cases. A two-dimensional linear subspace obtained by 2-dimensional PCA is informative for a number of datasets. As we intend to compare a large set of classifiers we restrict ourselves by some choices.

1. No feature scaling is included. This would be profitable for about 100 datasets, but it would double the set of experiments to be run with and without feature scaling. For other datasets feature scaling would also emphasize the artificially added noise.
2. Since many classifiers cannot be directly applied to the set of 10000 objects, the following scheme has been followed:
   (a) The 65 datasets with more than 500 objects are split at random in equally sized subsets smaller than 500.
   (b) For each of these subsets a version of the particular classifier is trained obtaining either posterior probabilities or class confidences in the [0,1] interval.
   (c) During classification these base classifiers are combined by the sum rule.
3. Class sizes can be very skewed. It is assumed that the observed class frequencies are representative for the test sets. So no corrections are made for the skewness.

4. Due to a large set of datasets we skip very time consuming snd advanced classifiers such as adaboost, neural networks and SVMs with kernels optimized by grid search.
5. Instead of a grid search we use 'smart' choices for the regularization parameter of the SVMs and the choice of the kernels. This will not directly influence our conclusions as we use the same SVM procedures in the feature space as well as in the dissimilarity space.
6. We also include one classifier in the 2D PCA space as good results are achieved in 2D subspaces for some datasets. This would not be an obvious choice for general problems but it seems appropriate in the setting of artificially generated datasets.

## 4 Experiments

We train all classifiers by 10-fold cross-validation. The total execution time is about five days. The results of our experiments are presented in Table 1. The classifiers are:

**1-NN,** the 1-nearest neighbor rule.

**k-NN,** the k-nearest neighbor rule. k is optimized by LOO (Leave-One-Out) cross-validation over the training set.

**ParzenC,** densities estimated by a single Parzen kernel. Its width is optimized by LOO cross-validation over the training set.

**ParzenD,** densities estimated by different Parzen kernels per class, using an ML-estimator for the kernel width. The variances of the kernels are for every dimensions proportional to the corresponding class variances.

**Nearest Mean,** the nearest mean classifier.

**UDA,** uncorrelated discriminant analysis assuming normally distributed classes with different diagonal covariance matrices. This routine is similar to the so-called Gaussian Naive Bayes rule.

**LDA,** linear discriminant analysis assuming normally distributed classes with a common covariance matrix for all classes.

**QDA,** quadratic discriminant analysis assuming normally distributed classes with different covariance matrices.

**Naive Bayes,** using histogram density estimates with 10 bins per feature.

**Logistic,** linear logistic classifier.

**FDS-0.2-Fish,** feature-based dissimilarity space classifier, using randomly selected 20% of the training set for representation and the Fisher discriminant for classification.

**FDS-Fish,** FDSC using the (pseudo-)Fisher discriminant for classification. For a complete dissimilarity space whose dimension is equal to the size of the training set, first the null-space is removed and then the linear classifier is constructed that perfectly separates the classes in the training set. This classifier is overtrained in comparison to a linear SVM as it uses all objects as 'support' objects.

**FDS-Logistic,** FDSC using the linear logistic classifier. Similarly to the (pseudo-) Fisher rule, this classifier is overtrained in the complete dissimilarity space.

**FDSC-C-SVM,** FDSC using the C-SVM rule to compute a linear classifier. The value of $C$ is set to 1 and is rather arbitrary.

**Table 1.** The averaged results per classifier: the mean classification error, the number of times the classifier is the best and the average classifier rank

| | Mean error | # Best Scores | Mean rank |
|---|---|---|---|
| 1-NN | 0.204 | 6.0 | 13.7 |
| k-NN | 0.165 | 12.5 | 8.6 |
| ParzenC | 0.172 | 13.0 | 9.6 |
| ParzenD | 0.209 | 6.0 | 12.7 |
| Nearest Mean | 0.361 | 1.0 | 16.8 |
| UDA | 0.168 | 43.0 | 10.4 |
| LDA | 0.202 | 16.5 | 9.6 |
| QDA | 0.216 | 1.0 | 12.0 |
| NaiveBayes | 0.162 | 30.0 | 9.3 |
| Logistic | 0.204 | 13.5 | 10.4 |
| FDS-0.2-Fish | 0.191 | 7.5 | 12.1 |
| FDS-Fish | 0.162 | 9.0 | 8.5 |
| FDS-Logistic | 0.157 | 11.0 | 7.8 |
| FDS-C-SVM | 0.170 | 13.0 | 8.1 |
| FDS-$\nu$-SVM | 0.159 | 8.0 | 7.1 |
| PCA2-FDS-Fish | 0.143 | 70.5 | 7.7 |
| C-SVM | 0.195 | 12.0 | 8.5 |
| $\nu$-SVM | 0.208 | 12.5 | 9.8 |
| RB-SVM | 0.160 | 15.0 | 7.3 |

**FDSC-$\nu$-SVM,** FDSC using the $\nu$-SVM rule to compute a linear classifier. $\nu$ is estimated from the class frequencies and the LOO 1-NN error. This error serves as an estimate of the number of support objects. It is corrected for the sometimes very skewed class frequencies.

**PCA2-FDSC-Fish,** the feature space is first reduced to two dimensions by PCA. This space is converted to a dissimilarity space, in which the (pseudo-)Fisher discriminant is computed.

**C-SVM,** C-SVM in a feature space with $C = 1$.

**$\nu$-SVM,** the $\nu$-SVM rule described above, now in a feature space.

**RB-SVM,** the radial basis SVM using an estimate for the radial basis function based on the Parzen kernel as found by ParzenC. As a 'smart' choice we use five times the width of the Parzen kernel as found by ParzenC and the $\nu$-SVM rule as described above.

All experiments are performed by PRTools [13]. The LIBSVM package is used for training SVM [14]. All classifiers in the dissimilarity space are linear, but they correspond to nonlinear classifiers in the original feature space thanks to the nonlinearity of Euclidean distance. All other classifiers are computed in the original feature space.

The best classifier, PCA2-FDSC-Fish, makes use of the analyst observation that a number of datasets is in fact just 2D. If we abstain from this classifier then still the dissimilarity-based classifiers perform very well, comparable or better than the radial basis SVM. A plausible explanation is that FDSC can be understood as a SVM with a variable kernel as illustrated in Section 2. It has, however, the disadvantage that the linear classifier in the dissimilarity space still depends on all objects and is not restricted to a set of support objects. It may thereby be outlier sensitive.

**Table 2.** Classification errors for most characteristic datasets. Best results per dataset are underlined.

| | D242 | D47 | D200 | D168 | D116 | D291 | D180 | D100 | D298 | D82 | D183 | D171 | D292 | D286 | D97 | D5 | D29 | D24 | D218 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1-NN | .120 | .328 | .083 | .250 | .350 | .321 | .049 | .235 | .611 | .302 | .074 | .166 | .049 | .101 | .164 | .530 | .526 | .416 | .060 |
| k-NN | .160 | .220 | .175 | .254 | .173 | .272 | .046 | .126 | .425 | .300 | .056 | .166 | .049 | .101 | .128 | .513 | .379 | .296 | .047 |
| ParzenC | .144 | .254 | .068 | .284 | .207 | .289 | .051 | .162 | .449 | .296 | .052 | .366 | .051 | .126 | .125 | .533 | .453 | .322 | .047 |
| ParzenD | .132 | .323 | .135 | .222 | .357 | .403 | .060 | .235 | .618 | .279 | .069 | .186 | .051 | .086 | .145 | .560 | .440 | .382 | .056 |
| Nearest Mean | .277 | .427 | .425 | .401 | .167 | .580 | .114 | .490 | .385 | .378 | .511 | .419 | .521 | .373 | .178 | .537 | .435 | .330 | .509 |
| UDA | .179 | .267 | .099 | .274 | .213 | .075 | .034 | .126 | .425 | .253 | .078 | .284 | .087 | .220 | .092 | .527 | .366 | .270 | .073 |
| LDA | .177 | .263 | .310 | .252 | .197 | .557 | .020 | .129 | .412 | .250 | .069 | .304 | .049 | .207 | .095 | .517 | .366 | .279 | .047 |
| QDA | .170 | .272 | .120 | .277 | .280 | .164 | .071 | .123 | .495 | .260 | .069 | .282 | .067 | .202 | .132 | .503 | .362 | .335 | .056 |
| NaiveBayes | .158 | .293 | .117 | .240 | .207 | .170 | .049 | .123 | .203 | .265 | .069 | .294 | .056 | .188 | .115 | .507 | .371 | .305 | .043 |
| Logistic | .172 | .263 | .304 | .254 | .197 | .554 | .034 | .129 | .409 | .236 | .069 | .301 | .054 | .220 | .099 | .513 | .379 | .279 | .052 |
| FDS-0.2-Fish | .219 | .310 | .175 | .285 | .277 | .216 | .034 | .132 | .498 | .281 | .048 | .337 | .067 | .200 | .102 | .550 | .440 | .352 | .047 |
| FDS-Fish | .158 | .289 | .182 | .307 | .217 | .167 | .031 | .126 | .462 | .289 | .056 | .133 | .041 | .086 | .095 | .533 | .466 | .365 | .047 |
| FDS-Logistic | .130 | .289 | .086 | .240 | .217 | .167 | .031 | .123 | .462 | .277 | .056 | .137 | .036 | .081 | .095 | .543 | .466 | .365 | .047 |
| FDS-C-SVM | .170 | .280 | .188 | .262 | .187 | .226 | .031 | .123 | .422 | .289 | .065 | .142 | .054 | .072 | .089 | .493 | .457 | .343 | .043 |
| FDS-$\nu$-SVM | .157 | .246 | .123 | .270 | .193 | .236 | .029 | .126 | .432 | .289 | .061 | .210 | .051 | .136 | .066 | .487 | .388 | .288 | .043 |
| PCA2-FDS-Fish | .158 | .319 | .093 | .302 | .240 | .197 | .040 | .123 | .302 | .352 | .061 | .161 | .044 | .099 | .086 | .120 | .427 | .403 | .060 |
| C-SVM | .200 | .250 | .286 | .242 | .187 | .430 | .029 | .129 | .412 | .248 | .061 | .308 | .056 | .202 | .086 | .517 | .332 | .292 | .043 |
| $\nu$-SVM | .200 | .254 | .286 | .267 | .193 | .528 | .037 | .179 | .415 | .272 | .061 | .335 | .056 | .244 | .092 | .490 | .366 | .249 | .043 |
| RB-SVM | .160 | .254 | .125 | .270 | .197 | .174 | .031 | .129 | .445 | .279 | .061 | .419 | .054 | .136 | .095 | .503 | .362 | .288 | .039 |

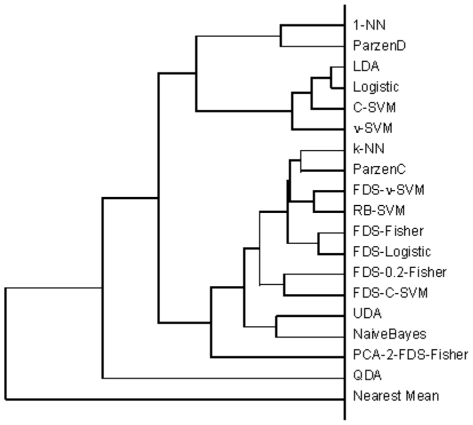

**Fig. 2.** Dendrogram

The classification of the test sets in the competition is based on the best classifier per dataset determined by the 10-fold cross-validation experiments. This classifier is trained by the entire training set. The performance of this classification rule can be estimated by the average of the minimum cross-validation error per dataset (of the best classifier for that dataset). This appears to be 0.106. This number is optimistically biased as it is based on the minima of 19 stochastic numbers. A better estimate would demand an additional 10-fold cross-validation loop over all classifiers. This would have increased the computation time to about 50 days. The meaning of such error estimates is of course very limited. It is an estimate of the expected error of the best classifier found by 10-fold cross-validation for a new dataset randomly selected out of the same distribution of datasets that generated the set used in contest.

We applied a cluster analysis to the $19 \times 301$ matrix of all classification errors in order to obtain a better view of the similarities of the various classifiers (as seen through the eyes of the set of contest datasets). In Figure 2 the dendrogram is shown resulting from a complete linkage cluster procedure. A number of obvious relations can be observed: k-NN and ParzenC, or the linear classifiers group (LDA, logistic and linear SVMs). The various FDS classifiers constitute a group with the RB-SVM, which make sense as all are nonlinear classifiers in the feature space.

## 5 Discussion

Our experiments are run on the set of contest datasets. The results show that the best linear classifiers in the dissimilarity space (FDSC-$\nu$-SVM) perform overall better than the linear as well as nonlinear SVM in the feature space. This result is of minor interest as it depends on the distribution of datasets in the contest. One should realize that for any classification rule, datasets can be either found or constructed for which this rule outperforms all other rules. As each classifier relies on assumptions, estimators or approximations, a dataset for which these are exactly fulfilled is an ideal dataset for that classifier.

**Table 3.** Rank correlations of the classification errors for the most characteristic datasets

| | 1-NN | k-NN | ParzenC | ParzenD | Nearest Mean | UDA | LDA | QDA | NaiveBayes | Logistic | FDS-0.2-Fish | FDS-Fish | FDS-Logistic | FDS-C-SVM | FDS-$\nu$-SVM | PCA2-FDS-Fish | C-SVM | $\nu$-SVM | RB-SVM |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1-NN | 1.0 | -0.0 | 0.3 | 0.8 | -0.4 | -0.3 | -0.5 | -0.1 | -0.0 | -0.5 | -0.3 | 0.3 | 0.7 | -0.1 | -0.2 | 0.3 | -0.7 | -0.7 | -0.4 |
| k-NN | -0.0 | 1.0 | 0.4 | -0.0 | 0.2 | -0.5 | -0.1 | -0.1 | -0.2 | -0.1 | -0.1 | -0.0 | -0.1 | 0.1 | 0.2 | -0.1 | -0.1 | 0.1 | -0.0 |
| ParzenC | 0.3 | 0.4 | 1.0 | 0.2 | -0.1 | -0.3 | -0.5 | -0.2 | -0.2 | -0.5 | 0.3 | 0.1 | 0.1 | -0.3 | 0.2 | -0.0 | -0.3 | -0.1 | 0.2 |
| ParzenD | 0.8 | -0.0 | 0.2 | 1.0 | -0.5 | -0.4 | -0.3 | -0.0 | 0.0 | -0.3 | -0.1 | 0.2 | 0.7 | -0.1 | -0.5 | -0.0 | -0.4 | -0.6 | -0.4 |
| Nearest Mean | -0.4 | 0.2 | -0.1 | -0.5 | 1.0 | 0.1 | 0.3 | -0.2 | 0.2 | 0.5 | -0.3 | -0.5 | -0.6 | 0.0 | -0.1 | -0.1 | 0.3 | 0.4 | 0.1 |
| UDA | -0.3 | -0.5 | -0.3 | -0.4 | 0.1 | 1.0 | 0.1 | 0.5 | 0.2 | 0.3 | 0.0 | -0.3 | -0.3 | -0.3 | -0.2 | -0.1 | 0.2 | 0.1 | 0.1 |
| LDA | -0.5 | -0.1 | -0.5 | -0.3 | 0.3 | 0.1 | 1.0 | -0.3 | -0.1 | 0.8 | -0.2 | -0.4 | -0.5 | -0.2 | 0.0 | -0.5 | 0.7 | 0.6 | 0.2 |
| QDA | -0.1 | -0.1 | -0.2 | -0.0 | -0.2 | 0.5 | -0.3 | 1.0 | 0.4 | -0.1 | 0.0 | -0.2 | -0.0 | -0.3 | -0.3 | 0.1 | -0.3 | -0.2 | 0.0 |
| NaiveBayes | -0.0 | -0.2 | -0.2 | 0.0 | 0.2 | 0.2 | -0.1 | 0.4 | 1.0 | 0.1 | -0.2 | -0.5 | -0.1 | -0.0 | -0.5 | -0.0 | -0.1 | -0.1 | -0.0 |
| Logistic | -0.5 | -0.1 | -0.5 | -0.3 | 0.5 | 0.3 | 0.8 | -0.1 | 0.1 | 1.0 | -0.4 | -0.6 | -0.6 | -0.2 | -0.1 | -0.4 | 0.6 | 0.7 | -0.0 |
| FDS-0.2-Fish | -0.3 | -0.1 | 0.3 | -0.1 | -0.3 | 0.0 | -0.2 | 0.0 | -0.2 | -0.4 | 1.0 | 0.3 | 0.2 | -0.1 | -0.0 | -0.2 | 0.1 | -0.0 | 0.3 |
| FDS-Fish | 0.3 | -0.0 | 0.1 | 0.2 | -0.5 | -0.3 | -0.4 | -0.2 | -0.5 | -0.6 | 0.3 | 1.0 | 0.7 | 0.3 | 0.1 | 0.4 | -0.5 | -0.6 | -0.3 |
| FDS-Logistic | 0.7 | -0.1 | 0.1 | 0.7 | -0.6 | -0.3 | -0.5 | -0.0 | -0.1 | -0.6 | 0.2 | 0.7 | 1.0 | 0.1 | -0.2 | 0.3 | -0.6 | -0.9 | -0.6 |
| FDS-C-SVM | -0.1 | 0.1 | -0.3 | -0.1 | 0.0 | -0.3 | -0.2 | -0.3 | -0.0 | -0.2 | -0.1 | 0.3 | 0.1 | 1.0 | 0.3 | 0.3 | -0.1 | -0.1 | -0.2 |
| FDS-$\nu$-SVM | -0.2 | 0.2 | 0.2 | -0.5 | -0.1 | -0.2 | 0.0 | -0.3 | -0.5 | -0.1 | -0.0 | 0.1 | -0.2 | 0.3 | 1.0 | 0.1 | 0.1 | 0.3 | 0.3 |
| PCA2-FDS-Fish | 0.3 | -0.1 | -0.0 | -0.0 | -0.1 | -0.1 | -0.5 | 0.1 | -0.0 | -0.4 | -0.2 | 0.4 | 0.3 | 0.3 | 0.1 | 1.0 | -0.6 | -0.4 | -0.4 |
| C-SVM | -0.7 | -0.1 | -0.3 | -0.4 | 0.3 | 0.2 | 0.7 | -0.3 | -0.1 | 0.6 | 0.1 | -0.5 | -0.6 | -0.1 | 0.1 | -0.6 | 1.0 | 0.7 | 0.4 |
| $\nu$-SVM | -0.7 | 0.1 | -0.1 | -0.6 | 0.4 | 0.1 | 0.6 | -0.2 | -0.1 | 0.7 | -0.0 | -0.6 | -0.9 | -0.1 | 0.3 | -0.4 | 0.7 | 1.0 | 0.5 |
| RB-SVM | -0.4 | -0.0 | 0.2 | -0.4 | 0.1 | 0.1 | 0.2 | 0.0 | -0.0 | -0.0 | 0.3 | -0.3 | -0.6 | -0.2 | 0.3 | -0.4 | 0.4 | 0.5 | 1.0 |

On the basis of the above it is of interest to observe that all classifiers that we studied here are the best ones for one or more datasets. This proves that the contest is sufficiently rich to show the variations in classification rules that we applied. Simple rules like Nearest Mean and 1-Nearest Neighbor are sometimes the best, as well as much more advanced rules like the Radial Basis SVM and the dissimilarity space classifiers.

To make the analysis less dependent on the accidental collection of problems in the contest, for every classifier we selected the dataset for which it is the best and for which the second best classifier is most different. This set of 19 datasets, one for each classifier, can be considered as a set of prototypical problems. Table 2 presents the 10-fold cross-validation errors for these prototypical datasets. Table 3 shows the rank correlations between the classifiers on the basis of the classification errors for these datasets.

In Table 2 the differences between the datasets can be clearly observed. A dataset that might be judged as very simple is D116, as Nearest Mean is the best. Dataset D291 is interesting as all linear classifiers fail and perform close to random, while UDA (Naive Gaussian) is very good. Dataset D29 shows an almost random performance for all classifiers and inspired us to include the two-dimensional subspace classifier PCA2-FDS-Fish. We were somewhat disappointed by our 'smart' choice for $\nu$ in the $\nu$-SVM classifier as it turned out to be very often worse than the C-SVM with the rather arbitrary choice of $C = 1$. This holds both for feature spaces as well as dissimilarity spaces.

The similarities and dissimilarities between the classifiers can be better judged from the rank correlations between the performances on the 19 prototypical datasets; see Table 3. Strong positive correlations indicate similar classifiers, e.g. 1-NN and ParzenD, LDA and Logistic or the two linear SVMs. Strong negative correlations can be observed between the linear and nonlinear classifiers, e.g. the FDS classifiers in the dissimilarity space. It is interesting that there is no correlation between the 1-NN and k-NN rules.

## 6 Conclusions

We have presented a set of classifiers based on a dissimilarity representation built on top of a feature representation. Linear classifiers in the dissimilarity space correspond to nonlinear classifiers in the feature space. The nonlinearity has not to be set by some kernel but results naturally from the object distances in the training set as they are used for representation. Consequently, there are no parameters to be defined if classification rules like the Fisher discriminant or the logistic classifier are applied. The contest shows a large set of examples for which this classification scheme outperforms traditional classifiers including linear and nonlinear SVMs.

**Acknowledgments.** We acknowledge financial support from the FET programme within the EU FP7, under the SIMBAD project (contract 213250) as well as the Engineering and Physical Sciences Research Council in the UK.

## References

1. Pękalska, E., Duin, R.: The Dissimilarity Representation for Pattern Recognition. Foundations and Applications. World Scientific, Singapore (2005)
2. Pękalska, E., Duin, R.: Beyond traditional kernels: Classification in two dissimilarity-based representation spaces. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 38(6), 729–744 (2008)
3. Pękalska, E., Duin, R.P.W.: Dissimilarity-based classification for vectorial representations. In: ICPR, vol. (3), pp. 137–140 (2006)
4. Edelman, S.: Representation and Recognition in Vision. MIT Press, Cambridge (1999)
5. Riesen, K., Bunke, H.: Graph classification based on vector space embedding. IJPRAI 23(6) (2009)
6. Xiao, B., Hancock, E.R., Wilson, R.C.: Geometric characterization and clustering of graphs using heat kernel embeddings. Image Vision Comput. 28(6), 1003–1021 (2010)
7. Fischer, A., Riesen, K., Bunke, H.: An experimental study of graph classification using prototype selection. In: ICPR, pp. 1–4 (2008)
8. Pękalska, E., Duin, R., Paclík, P.: Prototype selection for dissimilarity-based classifiers. Pattern Recognition 39(2), 189–208 (2006)
9. Jacobs, D., Weinshall, D., Gdalyahu, Y.: Classification with Non-Metric Distances: Image Retrieval and Class Representation. IEEE TPAMI 22(6), 583–600 (2000)
10. Duin, R., de Ridder, D., Tax, D.: Experiments with object based discriminant functions; a featureless approach to pattern recognition. Pattern Recognition Letters 18(11-13), 1159–1166 (1997)
11. Graepel, T., Herbrich, R., Bollmann-Sdorra, P., Obermayer, K.: Classification on pairwise proximity data. In: Advances in Neural Information System Processing, vol. 11, pp. 438–444 (1999)
12. Pękalska, E., Paclík, P., Duin, R.: A Generalized Kernel Approach to Dissimilarity Based Classification. J. of Machine Learning Research 2(2), 175–211 (2002)
13. Duin, R., Juszczak, P., de Ridder, D., Paclík, P., Pękalska, E., Tax, D.: PR-Tools (2004), `http://prtools.org`
14. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines (2001), Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`